Electronic Version

Stylesheet Version v1.1.1

Claims

[c1] 1. A robust method for updating a computer-stored hierarchical structure of nodes via a node identification technique, said update retaining properties and parent/child relationships of said hierarchical structure without renumbering existing node ID values associated with said hierarchical structure, said method comprising the steps of:

(a) receiving an instruction to insert a new node at an insertion point in said computer-stored hierarchical structure;

(b) identifying one of, or a combination of the following: a left node ID value closest to the left of said insertion point or a closest right node ID value closest to the right of said insertion point;

(c) calculating a new ID value based upon node ID value(s) identified in (b), said calculated value greater than ID values of nodes to the left of said insertion point and less than ID values of nodes to the right of said insertion point; and

(d) updating said computer-stored hierarchical structure by inserting said new node and associating said inserted node with said calculated ID value, wherein order, node

ID values, and relationships between parent, child, and siblings in said hierarchical structure of nodes remain unchanged with said insertion of new node.

[c2] 2. A method as per claim 1, wherein said new ID value is calculated via any of the following steps: concatenating said left node ID value with one or more high key values and a positive value, decreasing last digit of said right node ID value, increasing last digit of left node ID value, decreasing last digit of said right node ID value and concatenating a positive value, or concatenating said left node ID value with one or more zeros and a positive value.

[c3] 3. A method as per claim 1, wherein a digit in said calculated ID value has a negative value.

[c4] 4. A method as per claim 1, wherein counts between nodes in said hierarchical structure of nodes have a gap and said high key value equal to said gap value.

[c5] 5. A method as per claim 1, wherein said ID values are encoded and are byte comparable.

[c6] 6. A method as per claim 1, wherein said nodes are associated with a mark-up language based document.

[c7] 7. A method as per claim 6, wherein said mark-up based

language is XML.

[c8] 8. A method as per claim 1, wherein said method is implemented in conjunction with a relational database.

[c9] 9. An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which updates a computer-stored hierarchical structure of nodes via a node identification technique, said update retaining properties and parent/ child relationships of said hierarchical structure without renumbering existing node ID values associated with said hierarchical structure, said medium comprising:
(a) computer readable program code aiding in receiving an instruction to insert a new node at an insertion point in said computer-stored hierarchical structure;
(b) computer readable program code identifying one of, or a combination of the following: a left node ID value closest to the left of said insertion point or a closest right node ID value closest to the right of said insertion point;
(c) computer readable program code calculating a new ID value based upon node ID value(s) identified in (b), said calculated value greater than ID values of nodes to the left of said insertion point and less than ID values of nodes to the right of said insertion point; and
(d) computer readable program code updating said com-

puter-stored hierarchical structure by inserting said new node and associating said inserted node with said calculated ID value, wherein order, node ID values, and relationships between parent, child, and siblings in said hierarchical structure of nodes remain unchanged with said insertion of new node.

[c10] 10. An article of manufacture as per claim 9, wherein said new ID value is calculated via any of the following steps: concatenating said left node ID value with one or more high key values and a positive value, decreasing last digit of said right node ID value, increasing last digit of left node ID value, decreasing last digit of said right node ID value and concatenating a positive value, or concatenating said left node ID value with one or more zeros and a positive value.

[c11] 11. An article of manufacture as per claim 9, wherein said ID values are encoded and are byte comparable.

[c12] 12. An article of manufacture as per claim 9, wherein said nodes are associated with a mark-up language based document.

[c13] 13. An article of manufacture as per claim 12, wherein said mark-up based language is XML.

[c14] 14. An article of manufacture as per claim 9, wherein

said medium works in conjunction with a relational database.

[c15] 15. An article of manufacture as per claim 9, wherein counts between nodes in said hierarchical structure of nodes have a gap and said high key value equal to said gap value.

[c16] 16. An article of manufacture as per claim 9, wherein a digit in said calculated ID value has a negative value.

[c17] 17. A method for updating a computer-stored hierarchical structure of nodes without renumbering existing node ID values associated with said hierarchical structure, said method comprising the steps of:
(a) receiving an instruction to insert a new node at an insertion point in said computer-stored hierarchical structure;
(b) identifying one of, or a combination of the following: a left node ID value closest to the left of said insertion point or a closest right node ID value closest to the right of said insertion point;
(c) calculating a new ID value via one of the follows ways:
(i) when no nodes are present either to the right or left of said insertion point, assigning a new ID value based on a level associated with said insertion point, wherein a number other than zero starts a count for said level;

(ii) when no nodes are present to right of said insertion point, calculating a new ID value via any of the following steps: increasing last digit of said left node ID value or concatenating said left node ID value with one or more zeros and a positive value;
(iii) when no nodes are present to left of said insertion point, calculating a new ID value via any of the following steps: decreasing last digit of said right node ID value, concatenating said left node ID value with one or more high key value, or concatenating said left node ID value with one or more high key value and a positive value; or
(iv) when nodes are present to the left and right of said insertion point, calculating a new ID value via any of the following steps: concatenating said left node ID value with one or more high key values and a positive value, decreasing last digit of said right node ID value, or increasing last digit of left node ID value, or concatenating said left node ID value with one or more zeros and a positive value,
said calculated value greater than ID values of nodes to the left of said insertion point and less than ID values of nodes to the right of said insertion point; and
(d) updating said computer-stored hierarchical structure by inserting said new node and associating said inserted node with said calculated ID value, wherein order, node ID values, and relationships between parent, child, and

siblings in said hierarchical structure of nodes remain unchanged with said insertion of new node.

[c18] 18. A method as per claim 17, wherein a digit in said calculated ID value has a negative value.

[c19] 19. A method as per claim 17, wherein said ID values are encoded and are byte comparable.

[c20] 20. A method as per claim 17, wherein said nodes are associated with a mark-up language based document.

[c21] 21. A method as per claim 20, wherein said mark-up based language is XML.

[c22] 22. A method as per claim 17, wherein said method is implemented in conjunction with a relational database.